# Comparison of the Three Techniques

Original Text: "I visited Dr. Adams today. He was very helpful."

* Sentence Tokenization:

["I visited Dr. Adams today.", "He was very helpful."]

* Punkt Sentence Tokenizer:

["I visited Dr. Adams today.", "He was very helpful."]

* Regex Tokenizer

["I visited Dr.", "Adams today.", "He was very helpful."]

In this example, the first two methods worked correctly, but the regex method split the sentence wrong after "Dr."

# Observations

**Over-splitting:** Regex Tokenizer splits incorrectly on abbreviations like ‘Dr.’ .

**Under-splitting:** Regex Tokenizer may fail if the text has no clear punctuation marks.

**Abbreviations & Decimals:** Sentence Tokenization and Punkt Sentence Tokenizer handled correctly but Regex Tokenizer didn’t split correctly

**Surprising behaviors:** Regex Tokenizer treats every period (.) as the of a sentence, even when it part of an abbreviation or number.

# 3.Reflection

After testing all three methods, and see the results I will choose Sentence Tokenization for future work because it’s the best choice; it is accurate with abbreviations and decimal and it works well without extra customization.

Punkt Sentence Tokenizer works well and can be improved by training it on specific types of text, such as healthcare records.

Regex Tokenizer is quick and easy to use, but it often makes mistakes, so it is not a good choice for messy real-world text.